

GT4 @ Intel - Experiences

Ravi Subramaniam – Intel Corporation

Contributors: Ralf Ratering, Sunil Munjal, Xubo Zhang

February 15th, 2006

Agenda:

- Describe GT4 efforts
- Identify positive and not so positive experiences with GT4

Overview

GT use since late 2002 through GT3 to GT4

GT4 currently used in prototype activities

- Infrastructure (SO) and relevance to Intel products
- Rich programming environment (GT4 as a target environment for such a system)

Engaged with GT through Globus Consortium too

Globus Relevance: Leading open-source software base with wide adoption/interest

- Toolkit approach allowing “ala carte” services/code base
- Aggressive in migrating to service-oriented paradigm and relevant standards
- Demonstrated usage in demanding environments
- Encouraging support and interaction with GT team

Use Globus code as appropriate (not using all aspects of Globus or as intact GT “services”) to build SO Grid – augmented with code from many other sources

Service Oriented Grids/Infrastructure

GT4 primarily focused as enabler of Proof of Concepts

Globus not used in the gateway mode that is the widespread model (code deployed to all resources)

GRAM modifications and additions

- Extended fork interface with sophisticated job and container management – retain WS interface and user interactions
- Job discovery and management capability – jobs proxies expose service oriented manageability interfaces
- Checkpoint and restart, non-VM based migration (live migration with client support), monitoring, interactive and batch
- Automated management of licenses
- Extended RSL

MDS modifications and extensions

- To support SQL interface (automated persistent store management)
- Unified schema for resource representation (indexed in MDS) – CIM – anticipates WS-CIM
- Support of WBEM, CIMOMs, WDM (Windows)
- First order fault tolerance

GSI

- Automated authenticated host credential management – bootstrap and renewal
- Integrated with Kerberos/Active Directory infrastructure (user and role)

Fully automated GT install with management of distributed dependencies

- Service lifecycle management

Service builder environment

GPE Portability

GPE Client Framework and GridBeans

GT4 Client API
implementation

CGSP Client API
implementation

Unicore/GS Client API
implementation

Atomic Service Interfaces

SOAP, WSDL, WSRF, WS-Addressing, WS-Security, JSDL, etc.

GT4 Atomic Service
Implementation

CGSP Atomic Service
Implementation

Atomic Service
Implementation

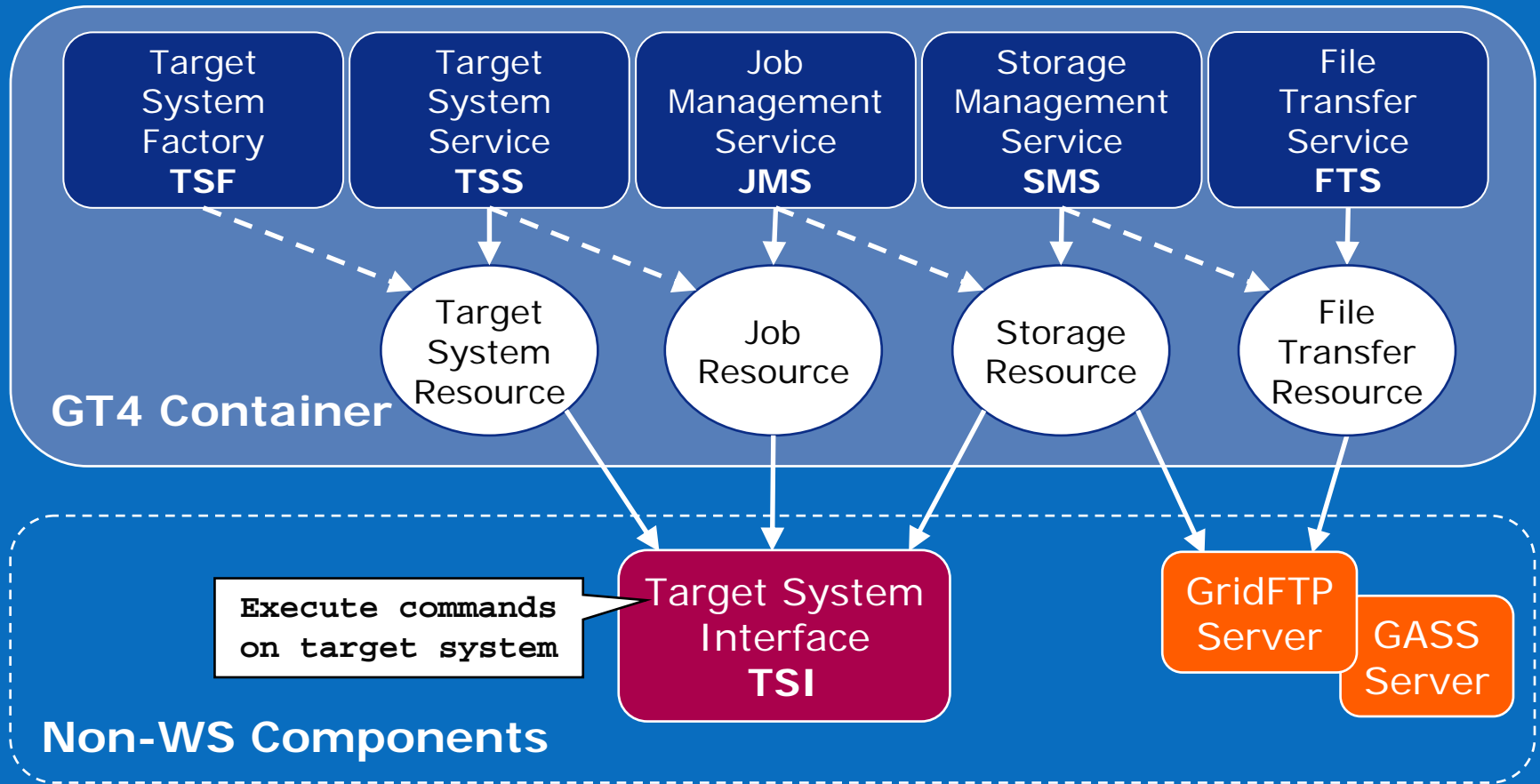
Globus Toolkit 4

China Grid
Supporting
Platform (CGSP)

Unicore/GS

- GPE Clients work across different infrastructures
- Unicore/GS provides native atomic service interfaces
- GT4 and CGSP add additional atomic services

Atomic Services for Globus Toolkit 4



- Target System Interface incarnates abstract operations to OS-specific commands

Positive Experiences

Aggressive Implementation of Standards

- Supports established and emerging standards (reference implementation)

Wide deployment

- Availability of code additions and other GT improvements
- Wider testing of implemented capabilities

Feature-rich service container

- Full WSRF implementation
- Good programming model
- C- Container
 - Light weight
 - Efficient SOAP stack and WSRF implementation

Good security infrastructure

- Proxy model suitable for problem of trust delegation
- Good implementation of PKI and supporting infrastructure

Good documentation

- Globus documentation project

Support for virtual workspaces

- New capabilities for dynamic resource provisioning and management
- Stronger security through encapsulation in partitions
- Hardware support with Intel Virtualization Technology (VT)

Elegant Aggregation framework in java

- Would be nice to have it in C too – hint, hint ..

Data Transport implementation

Areas for improvement(?)

Frequent changes in the architecture. Not production status yet

GT-4 Java container

- Rather heavy weight container to be deployed on every machine
- Limited number of simultaneous connections
- Slow compared to other implementation
- Difficult to deploy and patch over 10,000 machines
- No mechanisms to manage lifecycle of services.
- No code security.
- Support for only PKI security and not Kerberos

C – Container

- Does not support dynamic management of services
- Problems with deserializing arrays as parameters in container (Workaround is simple changes in service stub that is generated)
- Poor support for aggregation framework
- Lacks container registry service
- WS Notification not fully implemented

Areas for improvement (contd.)

GRAM

- No way to customize for pre/post execution steps.
- GRAM is primarily designed for submitting jobs to cluster and not for forking and controlling jobs.

MDS

- Scalability – registry updates cannot handle more than few hundreds of registrants
- I/O performance – use of java nio suggested
- Registry content (XML doc) size seems to be limited to a few MB – bug or design flaw?

Old WSRF version

- GT4 uses WSRF draft 2004/06
- No interoperability with current WSRF implementations like UNICORE/GS

Proprietary service interfaces

- Agreement on standardized service interfaces in GT4 would be desirable (OGSA-Life?)
- Atomic service interfaces could be starting point
- Need more functionality in GRAM, e.g. abstraction/incarnation mechanism

Focus on scientific community, less established in enterprise world

- How can we take advantage of GT4 in service-oriented business solutions?
- Integration of GT4 services into GPE-BPEL workflows as first step